Đã bắt đầu vào lúc	Thứ sáu, 7 Tháng mười 2022, 9:53 AM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ bảy, 15 Tháng mười 2022, 10:02 PM
Thời gian thực hiện	8 ngày 12 giờ
Điểm	10,00/12,00
Điểm	**8,33** của 10,00 (**83,33**%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

Given a positive number, print following a pattern without using any loop.

```
Input: n = 16
Output: 16, 11, 6, 1, -4, 1, 6, 11, 16 (has no space at the end)

Input: n = 10
Output: 10, 5, 0, 5, 10 (has no space at the end)
```

We basically first reduce 5 one by one until we reach a negative or 0. After we reach 0 or negative, we one add 5 until we reach n.

Note: Please note that you can't using key work for, while, goto (even in variable names, comment).

You can implement other recursive functions if needed.

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`printPattern(14);`	`14 9 4 -1 4 9 14`

Answer: (penalty regime: 0 %)

Reset answer

```
1  void printPatternRecursion(int n, bool up, int tag){
2      if(up && n == tag) {
3          cout<<n;
4          return;
5      }
6      cout<<n<<" ";
7      if(up == false){
8          if(n-5 <= 0) printPatternRecursion(n-5,true,tag);
9          else printPatternRecursion(n-5,false,tag);
10     }
11     else{
12         printPatternRecursion(n+5,true,tag);
13     }
14 }
15 void printPattern(int n)
16 {
17     /*
18      * STUDENT ANSWER
19      */
20     printPatternRecursion(n,false,n);
21 }
22
```

	Test	Expected	Got	
✔	`printPattern(14);`	`14 9 4 -1 4 9 14`	`14 9 4 -1 4 9 14`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Không chính xác

Điểm 0,00 của 1,00

Implement function

```
void printArray(int n){}
```

to print 0, 1, 2, ..., n (n is positive integer and has no space at the end).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`printArray(5);`	`0, 1, 2, 3, 4, 5`
`printArray(10);`	`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10`

Answer: (penalty regime: 0, 0, 0, 5, 10, 15, ... %)

Reset answer

```
1  void printArray(int n)
2  {
3      if(n==0){
4          cout<<n;
5          return;
6      }
7      printArray(n-1);
8      cout<<", "<<n;
9  }
```


Please provide a non-empty answer

Kiểm tra

Failed to run tests

Broken question (missing or duplicate prototype 'cpp_function_recursion_check'). Cannot be run.

Bài nộp không hợp lệ, và đã được bỏ qua mà không bị trừ điểm.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

Implement function

```
int findMax(int* arr, int length){}
```

to find the largest element using recursion (with length is the number of elements in integer array arr).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`int arr[] = {10, 5, 7, 9, 15, 6, 11, 8, 12, 2};` `cout << findMax(arr, 10);`	15

Answer: (penalty regime: 0, 0, 0, 5, 10, ... %)

Reset answer

```
1
2  int findMax(int* arr, int length)
3  {
4      /*
5       * STUDENT ANSWER
6       */
7      if(length-1==0) return *arr;
8      int max = findMax(arr+1,length-1);
9      if(*arr>max) return *arr;
10     else return max;
11 }
```

	Test	Expected	Got	
✔	`int arr[] = {10, 5, 7, 9, 15, 6, 11, 8, 12, 2};` `cout << findMax(arr, 10);`	15	15	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 4

Chính xác

Điểm 1,00 của 1,00

Implement function

```
bool isPalindrome(string str){}
```

to check if the given non empty string is palindrome, else not palindrome using recursion.

In test case, for extra point, we will have some palindrome sentences (All remaining test cases are words).

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream>, #include <string.h> and using namespace std;

For example:

Test	Result
`cout << isPalindrome("mom");`	1
`cout << isPalindrome("do geese see god");`	1

Answer: (penalty regime: 0 %)

Reset answer

```
bool isPalindrome1(string str){
    int len = str.length();
    if(len <= 1) return true;
    if(str[0] == ' ' && str[len-1] != ' '){
        if(len-2==0) return true;
        string sub = str.substr(1,len-1);
        return isPalindrome1(sub);
    }
    if(str[0] != ' ' && str[len-1] == ' '){
        if(len-2==0) return true;
        string sub = str.substr(0,len-1);
        return isPalindrome1(sub);
    }
    if(str[0] == str[len - 1]) {
        if(len - 2 == 0) return true;
        string sub = str.substr(1, len-2);
        return isPalindrome1(sub);
    }
    else return false;
}
bool isPalindrome(string str){

    return isPalindrome1( str);
}
```

	Test	Expected	Got	
✔	`cout << isPalindrome("mom");`	1	1	✔
✔	`cout << isPalindrome("do geese see god");`	1	1	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 5

Chính xác

Điểm 1,00 của 1,00

Give two positive integers a and b, implement function

```
int findGCD(int a, int b){}
```

to find **GCD** (Greatest Common Divisor) of a and b using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`cout << findGCD(124,32);`	4

Answer: (penalty regime: 0 %)

Reset answer

```
int findGCD(int a, int b)
{
    if(b==0) return a;
    return findGCD(b,a%b);
}
```

	Test	Expected	Got	
✔	`cout << findGCD(124,32);`	4	4	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 6

Chính xác

Điểm 1,00 của 1,00

Give a positive integer x, implement recursive function

```
void printHailstone(int number){}
```

to print the Hailstone Sequence of a given number upto 1 (no space at the end).

Hailstone Sequences follow these rules:

- If a number is even, divide it by 2
- If a number is odd, multiply it by 3 and add 1.

Example:

```
If number = 5. 5 is odd number so next number is 5*3 + 1 = 16. 16 is even number so next number is 16/2 = 8...
Finally, we get Hailstone sequence: 5 16 8 4 2 1.
```

You can find more information at: https://diendantoanhoc.net/topic/89145-d%C3%A3y-s%E1%BB%91-hailstone/

Note: Please note that you can't using key work for, while, goto (even in variable names, comment).

You can implement other recursive functions if needed.

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`printHailstone(32);`	`32 16 8 4 2 1`

Answer: (penalty regime: 0 %)

Reset answer

```
1  void printHailstone(int number)
2  {
3      /*
4       * STUDENT ANSWER
5       */
6       cout<<number;
7       if(number == 1) {
8           //cout<<number;
9           return;
10      }
11      cout<<" ";
12      if(number%2) printHailstone(number*3+1);
13      else printHailstone(number/2);
14 }
```

	Test	Expected	Got	
✔	`printHailstone(32);`	`32 16 8 4 2 1`	`32 16 8 4 2 1`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 7

Chính xác

Điểm 1,00 của 1,00

Function

```
int myArrayToInt(char* str, int n){}
```

takes a **string str** (which represents an positive decimal number), **n** is the number of elements in the string as arguments and returns its value.

Please note that you can't using key work for, while, goto (even in variable names, comment)

For this exercise, we have #include <iostream>, #include <string.h> and using namespace std;

For example:

Test	Result
`char str[] = "2020";` `printf("%d", myArrayToInt(str, 4));`	2020

Answer: (penalty regime: 0 %)

Reset answer

```
int myArrayToInt(char *str, int n)
{
    /*
     * STUDENT ANSWER
     */
    if(n==1) return str[0]-48;
    return myArrayToInt(str,n-1)*10 + (str[n-1]-48);
}
```

	Test	Expected	Got	
✔	`char str[] = "2020";` `printf("%d", myArrayToInt(str, 4));`	2020	2020	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 8

Chính xác

Điểm 1,00 của 1,00

Give two positive integers a and b, implement function

```
int findLCM(int a, int b){}
```

to find **LCM** (Lowest Common Multiple) of a and b using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`cout << findLCM(10, 102);`	`510`

Answer: (penalty regime: 0 %)

Reset answer

```
int findGCD(int a, int b)
{
    if(b==0) return a;
    return findGCD(b,a%b);
}
int findLCM(int a, int b)
{
    int GCD = findGCD(a,b);
    return (a*b)/GCD;
}
```

	Test	Expected	Got	
✔	`cout << findLCM(10, 102);`	510	510	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 9

Không chính xác

Điểm 0,00 của 1,00

Given a string, implement function

```
int strLen(char* str){}
```

to calculate length of the string using recursion.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

For example:

Test	Result
`char str[] = "Truong DH Bach Khoa";` `cout << strLen(str);`	19

Answer: (penalty regime: 0 %)

Reset answer

```
int strLen(char* str)
{
    /*
     * STUDENT ANSWER
     */
     if(*str == '\0') return 0;
     else return 1 + strlen(str+1);
}
```


Please provide a non-empty answer

Kiểm tra

Failed to run tests

Broken question (missing or duplicate prototype 'cpp_function_recursion_check-1-4'). Cannot be run.

Bài nộp không hợp lệ, và đã được bỏ qua mà không bị trừ điểm.

Câu hỏi 10

Chính xác

Điểm 1,00 của 1,00

Given a string `s` representing a sentence consisting only of `a-z` and `A-Z` and space character.
Your task is to implement a function with following prototype:

```
string reverseSentence(string s);
```

The function returns the reverse sentence of sentence `s`.
The testcases ensure that there is only one space character between two adjacent words, and the sentences do not begin or end with any space characters.

Note:
- The `iostream` library has been used and `namespace std` is being used. No other libraries are allowed.
- Using loop keywords (`for`, `while`, `do`) are not allowed, even in comments and variable names.
- You can write helper functions.

For example:

Test	Result
`cout << reverseSentence("data structure and algorithm is scary");`	`scary is algorithm and structure data`

Answer: (penalty regime: 0, 0, 0, 5, 10, 15, ... %)

Reset answer

```
string pickW(string s, int size, int index) {
    if (index >= size) return "";
    if (s[index] != ' ') return (s[index] + pickW(s, size, index + 1));
    else return "";
}

string reverseSentenceRecursion(string s, int size, int index) {
    string a = pickW(s, size, index);
    int b = a.size();
    if ((index + b + 1) >= size) return a;
    return reverseSentenceRecursion(s, size, index + a.size() + 1) + " " + a;

}
string reverseSentence(string s) {
    return reverseSentenceRecursion(s,s.size(),0);

}
```

	Test	Expected	Got	
✔	`cout << reverseSentence("data structure and algorithm is scary");`	`scary is algorithm and structure data`	`scary is algorithm and structure data`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 11

Chính xác

Điểm 1,00 của 1,00

String s contains lowercase letters, digits, "(" and ")", satisfying the following rules:

- Two digits cannot be adjacent.
- Two "(" cannot be adjacent.
- One "(" and one ")" cannot be adjacent.
- After any digit, there must be "(".
- The quantities of "(" and ")" are equal.

Change string s until new string t created, t contains only lowercase letters. These are changing rules:

- Sub-strings with form "n(p)", can change to "pp...p" (n times p), where n is a digit and p is a string.
- If p still contains "(", ")" or digits, continue to implement the above changing method.

Request: Implement function

```
expand(string s);
```

Where `s` is a string with the above form; return the result is a string containing only lowercase letters.

Example:

- String "2(ab3(cde)x)" changes into "abcdecdecdexabcdecdecdex".
- String "2(x0👆)3(z)" changes into "xxzzz".

Note: In this exercise, libraries `iostream`, `string` and `using namespace std;` have been used. You can add other functions for your answer, but you are not allowed to add other libraries.

For example:

Test	Result
`cout << expand("2(ab3(cde)x)") << "\n";`	abcdecdecdexabcdecdecdex
`cout << expand("2(x0(y))3(z)") << "\n";`	xxzzz

Answer: (penalty regime: 0 %)

Reset answer

```
string repeatString(string s, int x) {
    if (x == 0) return "";
    else return (s + repeatString(s, x - 1));
}
int findNumber(string s, int index) {
    int lenS = s.length();
    if (index == lenS) return -1;
    if (s[index] >= '0' && s[index] <= '9') return index;
    return findNumber(s, index + 1);
}
int findRepeatS(string s, int indexS, int numS, int numE) {
    if (numS == numE) return indexS - 1;
    if (s[indexS] == '(') return findRepeatS(s, indexS + 1, numS + 1, numE);
    if(s[indexS]==')') return findRepeatS(s, indexS + 1, numS , numE+1);
    return findRepeatS(s, indexS + 1, numS, numE);
}
string expandRecursion(string s) {
    if (findNumber(s, 0) == -1) return s;
    int indexS = s.find("(");
    int lenS = s.length();
```

```
    int numRe = s[indexS - 1] - '0';
    int indexE = findRepeatS(s, indexS+1,1,0);
    string sub = s.substr(indexS + 1, indexE - indexS - 1);
    if (indexS == 1 && indexE < lenS - 1) {
        return repeatString(expandRecursion(sub), numRe) + expandRecursion(s.substr(indexE + 1, lenS - indexE
    }
    if (indexS > 1 && indexE < lenS - 1) {
        return s.substr(0, indexS - 1) +
            repeatString(expandRecursion(sub), numRe) +
            expandRecursion(s.substr(indexE + 1, lenS - indexE - 1));
    }
    if (indexS > 1 && indexE == lenS - 1)
        return s.substr(0, indexS - 1) +
        repeatString(expandRecursion(sub), numRe);
    return repeatString(expandRecursion(sub), numRe);
}
string expand(string s) {
    return expandRecursion(s);
}
```

	Test	Expected	Got	
✔	`cout << expand("2(ab3(cde)x)") << "\n";`	abcdecdecdexabcdecdecdex	abcdecdecdexabcdecdecdex	✔
✔	`cout << expand("2(x0(y))3(z)") << "\n";`	xxzzz	xxzzz	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 12

Chính xác

Điểm 1,00 của 1,00

Given a string `s` consisting only of `'('` and `')'`.
Your task is to implement a function with following prototype:

```
int mininumBracketAdd(string s);
```

The function returns the mininum number of brackets needed to be inserted to `s` so that the brackets are balanced.

More info:
A sequence of brackets is balanced when there are no unmatched brackets.
Example: `()(())` is balanced, but `))()` is not.

Note:
- The `iostream` library has been used and `namespace std` is being used. No other libraries are allowed.
- Using loop keywords (`for`, `while`, `do`) are not allowed, even in comments and variable names.
- You can write helper functions.

For example:

Test	Result
`cout << mininumBracketAdd("))(((");`	5

Answer: (penalty regime: 0 %)

Reset answer

```
int mininumBracketAddRe(string* s, int numberOfStart, int addition, int index){
    if((*s)[index]=='(') return mininumBracketAddRe(s,numberOfStart+1,addition,index+1);
    if((*s)[index]==')') {
        if(numberOfStart) return mininumBracketAddRe(s,numberOfStart-1,addition,index+1);
        else return mininumBracketAddRe(s,numberOfStart,addition+1,index+1);
    }
    return (numberOfStart+addition);
}

int mininumBracketAdd(string s) {
    string* refS = &(s);
    return mininumBracketAddRe(refS, 0, 0, 0);
}
```

	Test	Expected	Got	
✔	`cout << mininumBracketAdd("))(((");`	5	5	✔
✔	`cout << mininumBracketAdd("))())))()()");`	4	4	✔
✔	`cout << mininumBracketAdd("");`	0	0	✔
✔	`cout << mininumBracketAdd(")()))())))()))))))(");`	12	12	✔

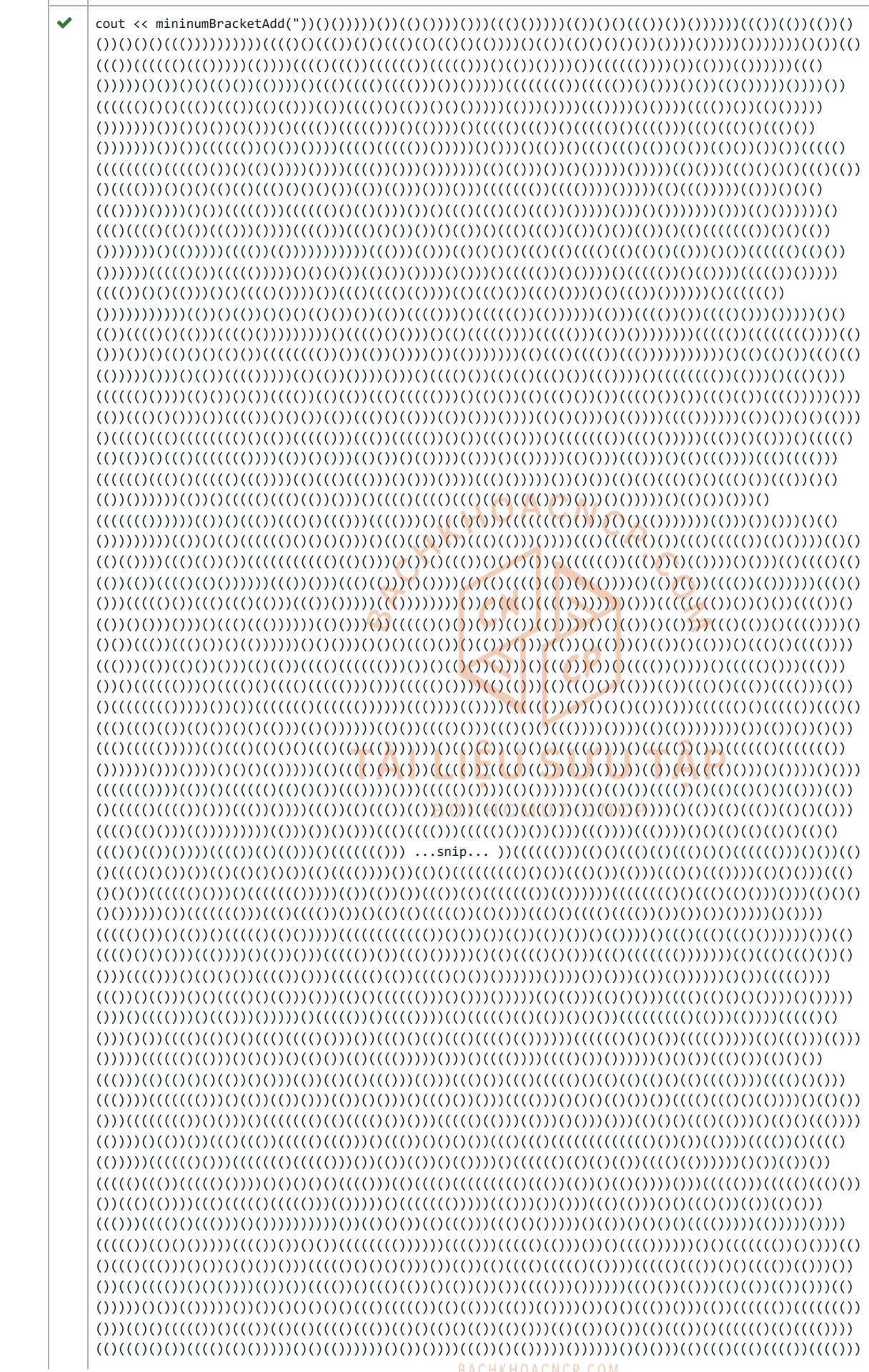
	Test	Expected	Got	
✔	`cout << mininumBracketAdd(")())((((())()())()()))()()))(()((((()(()()()))()((())((()))))))()((((()` `(((())(((()()(())(())))))(()))");`	10	10	✔

✔	`cout << mininumBracketAdd(")(()())()))))((((())))))()))(((())(((()((((())((()(()))())(()((()`
	`((((((())))(((()))))((())((()()))((())(()((((()())()()(()))()))(())))((((()()(()))()))))()()()`
	`(()))()))()())()(()()))))())())())())())((()()(())((()((((((((()()()())(()()()))()()()())`
	`()(()(())())((()))((((()((((((()((()))))))()))((()()))()()(((((()()()(()())((((((((()))(()))))`
	`(()))))()())())(((())()())()(((()))(())(((()((((())))((((()(()((((()((()(())))))))()(()(())()())`
	`((((()(()(())()()())()(()())))(((()))()())()()))))((()())(())()))(()))())))()())()()(()()((((((()(()`
	`()(()()))(()(()(())()(((((()(((()(()(()))))()())((()()(((()))(()))()(())()((()()(((((()((((())(()())`
	`())))()((()))((()))(()())()))()()))()))(()((()()()((()))(())(())((((())()())(())()((())((((()(())`
	`(()))(())()((((())((()())((()))((((((()(())()()()()(()(()))(()()())))))()))))()()))(()((((((((((()))`
	`(()))(()))(()))(()((()()((((())))(()(())((((()))(())))))(()((()((())))()))(()(((())()))))(()))(()`
	`()))()((((()((((()()(()))()((()(((())");`
		70	70	✔

	Test	Expected	Got	
✔	cout << mininumBracketAdd(")()))))))()())(((((()((()()))()((()()(((()((()))()))())())()))()(()()))(((() ()(((())))()))()((()))()))))()((()(((((((()()(()))()()))))))(()))((()(()))))((()(())()()(((())(()) ()()))((((((((()()(())()()((((()(()))(()()))))((())()())()(()))()))((((()((())(()))(((()()() (())(()()(()))(((((()))()((()()))))(()((())()()()))()))((()((((()()()()(()()()(())(()(())()()))(((((() ())))))(())(()(()(((((()()(((()(()))()))((()()(()(())))))(())))()()(()))()((()()(((((())(()()(() ()())(()))))(((((()(()())()((()()))()(())())(()))))))(()(())(()()((((((((()((()(((())(((()(())((() ((()))()())((()()((((()()())(()))()))(()((()))))(((()()))(((((()((()((()()())(()()())()()))()()))))() ()())(()()))()))()(()()()()((()))))()()()()(()()(((()()()))))()()()((()((((((())(((()(())()() ()((()()))(())()(()()()()(((())))(((((()()()()())((()))))()()))(()()))())(((()()(((((()()() ((((((()((()(()((((()()()()()((((()))()()(()(()))(()()))()(()(((()()()()(((()()(()))(((()))((()() (((((()))()((((()))(((()(((()((((()()((())))((()()()((()()(()(((()()))))(())())()(()))()))()(())() (()((()))()))()))())((((((())))))((()())()((((()(()(((()(()((()())))()(()((())()(()))()))())() (()()()((((((()(())()()(()))()())()))()))))()()))())((()((()((()))())))((()()))(((((())()))()((()))()) ()))(()))((((()((()()()())))))(()))(()(((()((())()(()((()(((()(()))((())((((()()()(()())() ((((()))((((()()()()(()())))))()())()()(((()()()()(())()(())()(()()()(())()())()()))(((()((((((()((())))(() ()(()))))()))()))(())()()((((((((()(())()))))()()))()()((()))))()()))()))((()(()()()(()()())) (())()()()()()))()()()(((((((()()()()()()(()())()()))()()))(()()))))))(()))()))(())())((((((() ()))))))()())()((((((((()(()))))()()))((((((()(()((((()(()(((()()()()(()()))))))()()()(((((()()))))) ((((()))(((((((()(((((()()(()))))()()()())))))))))()())(())))()(()))(())((((()))(())()))()((((()))) ((((()))()))(())()()()(()()((((()()()))))()))(()()())))))(()()()(())(()())()((((()(()(((()()(()) (())))(((()(()((((((()()(())((()))()()()()))()(()()(()((((((()()()(())()))()(()()))))()))(())()() ()()()(()))()(()((((((()())()()(()()()()))(()()()()(()))()()())()(()()()(()()()()(()(()()()))()(()))()) (()(())((((()))))(((()((()()(()))()()(()()()()(()))()()(()))()()()(()()(())()(()()())))))()(()))))(())) ((((((()()(((((((()))()))((((((()()()))())))))(((()(((((((()(()((()()(()()(()()()))(()()()(()(()))()()() ()))()()()(((()((()))))()((()()))((()()))(((()(((()()))()))()())))((()()()()(())(()()()(((()) (()(((((()())()))))())()()()()()())))))((()(()))(())))()()))()()()(()()()(((()()))(())()()()))() (((((((()())()()()()((())()()())())()))((((()()()()(()))))(())()()()()())(()((((()))()()()()(((()))() ()())()()()()))(((((((())))))()()(()()((((()))(((()()(())))(()))()()(()())))(((()))()(()(()) (()(((((())))))()()(()()(()()(()()()()))()()()()))())((((((()()((()()()(((()()()())(((((()()())(())))) ((()()()())(((())))(((()()))()(((()(((())))))))()((((()()()()()()()()()(((()()()(()(((()()()()))())()() ())(()((()()))))))()((((((()()())(((((()()())()(((()((((()((()))(()())()(()))))))()(())()()()()() (())())()(()))))())))((()(()(()()()()))))((())((()(()(()(((((()()()(((()()())(((((((()())())())() (()(())())))()()))()))((((((()(((()))()(()())()((((()(((())(()(()((())))()(()())()))()(((()()))) ((((((()))))((((()()()()(()))(()()()()))((()()()()((()()(()))((((())))))))(((()()()()))))()((((())())))()() (()))))((()(())())()()()(()()()))()(()()()()((()()()))(()(()))(()()(()(()()())(()((()()()))))(())()())))()()))(()() (()()(()())()))(()())(()))((((())))(((((()))()(((()())))()(((((()))))((((()()(((((()()(((()(((())()())(()))(()(()))))(()(()))(((((()(((()))(((((((()))))()((((()(((()())()()((((((()()()()) (()(()))()()))))(())()(()()(()))))))(()()))()((((()(((()))))(()()()()()(((((()()))))()()(())(()) ()())(((((())((()()()()()()())(((()))((((((((()()()()((((()))(((()()))()((((()((((()()() ()(())()(()()))()((((((()))))))((()()()()()()()()()(()))))()(((()()(((((()(())))))))(())(()() ())))(((()((((((((())(((() ...snip... ()))()((())()))()()((((()()()(()))((()())()()))())()))(()) ()))))))((((((()()(()(()())(()))()()))()())(((((((()((()())()()()))()((((()()(()())))) ((()))(((()()()(()))()()()()()()(()()()((()((()))(()()(()))()()(())())((((()())(()(((((()())()(() ()))))((((())(()((((((()(()())()(()))))(((())()))(()()))(()))()()(()(()((())()())()(()))))(()) ((((()()()))(((()((()))))(())(((())(((((()()(((()(()))()))())(()()(()(((((()())))()(()()()()))()) (())(((()()(((()))()(())((()())((((((((()()((((()()(((()()()))((((()()(((()()((((()()(((()()(((((() ()))(((()()()))))()()()(())((()(()()))))(((()))))(((()()()())()()())(())(()(((()((((()))))() ((((((((((()()))))))()()(())(()()()()()()()()(())((((((()()(()(((((((()(((()()(())(())(()()())) ()(()))))))(()))()))(()))(()((()))((((()()(()(((((()(((()))))(()()()()))()(()))((((())))))(()))(()) ())(()(((()()(()())))))))())((((()(((()()()(()(()(()()))(()()()(()))(((()()()()))(()(((((() (((((()(((()()(()(((()()()(()(((()())))))()))))(((()))))()())()))))))(((()(((()()((((()()()(((() (()(()))()()((((()()()()((()(((())((((((((((()))))))))()()()()()(()()))))(((((()(((()))))))))(())))) ()))(()()()()())))())))()(()()())()))))()()()()(()(()())()()()((()()(((((()))((()((()((()() (((()))()())(((((()()()()()(()()(((()()((((()()()))))(()()))()(()(((((()(((()()((((((()(((((()())) (()()))))()())((()()()()((()()(()))))(((()()(((((()))))))()))(()((((())))((()(((((()()()(((((() (()())(())((()()))))()((()))))()()()(()(((((()((((()()()()(()()((((((()))()()()))((()((()()((() (())((((()(((()()((()()()((()(()())))())))())()(()()()))()()()()()())))()(()))))(((((()()()))()()()(((() ((((()()()))()(()()()(())(((())()())()))())))))((()()()()((((()))(((()()(())))(((()))(((()()() ((((()()))((((()()()))))))))()((((()))))))))))()(()))()())(((()(()(()(((((()))(()((((((()(() (((((((((()()()()(()((()))((()()(()))(()((()))()(((()))()((((((()))))(((()))(((()())(((((()))))(())((((() ((((((()))))))()((((()))()()((()()))()()))((()()))()()(())(()()()())))(((()))((()))(((((() ((()()()()))()(((((()))()()(((((()()))()(()()()((((((()()()(()()()(()()()()(((()()((((((()()(() ())((()))()(((()((((()()))())))()((((()()((((()))))()((())))(((()())))))(((()()()))()()()()()() ((((())))()((((()((((((())))()(()(((((()())))(()()(()()()))(()(((((()()))(()()())))))()()(((()()(() (()())(()))))(()((()())()()(((())())))))()(()()(()())()()((((()())))()())()())()(())))))))()((((())	82	82	✔

	Test	Expected	Got	
	())()()((((((())))()))))()))((()))()(()))(())()))((()))()))((())((()(()))()()()))))(()(()))))(())) (((((()(()()))()((()))((())((())((()(())())(()(((()())(()())(()()())()()((((((()(()((()))())(()))(() ())((())())((()()()())))()))(((()))))()))((()()(((()(()((()))((()()()))((()()))()))))(()(()())(() ((())())(()((())))))))(())))))())((()))()((()(()((((((()(()())(()())((())(())()))))())())(()() (())())(()()((()))(()())(())(())()())((((()())()()()))))(()))))()(()))((()))((()((())()((((()((() (((((()((())()()()))))()()(()))()()()())(()()))(())(()(()((()))))(()())(()((((())))))((())((((()())) ())())())(((((()()())(()))())))((()((()())(((((())))()))()(()((())((()()))((())()()()(()))())()((() ()(())())()())()))))())()()))()))())(()()))))()))((()(())((()())(()()(()(((()(()(()))((((((())(())) ()(()((()((()()())()()((()())(()(()()))((((()))()()(())(()))))(())(()(()()(((((()((((()((() ((((((((((()))))()))(()(()(())((()))(()))())(())()()(()())()))()(()))()()((()(())(())()))))))()))() ((()(()))()))()())()(((((()())(()())(((((()()()()))((()))())())()))))())))())(())()))))))(()()()) ()))())())()(()()(()()))(()))(()())((((()())()()))())()((((()))(()(()())()()(()()))()())())())))) (()))(((((((()(())())()))(()())))))(((((()()())()))())()))(()))(())(()))(()()()(((((())()(()(((((()) ())))(()((()))(()(()()))()(()()))))()((()(()())()))()())))(())()(())((())(()())()()()))))())()(()) (((((()()()))((((())())))())))))(()((()(()(()))))()())()))()))))))())()(()(())(()))))(((()()()))))()) (((()()(()())(())())((()(()(()))))(()((()()()))))()))(((((()))))((((((())()(())((((())())");			

	Test	Expected	Got	
✔	`cout << mininumBracketAdd("))()()))))))())(()()))))))(((()))))))(()()()(((()))))))))((()))(((()))()()`	278	278	✔



	Test	Expected	Got	
	()))))))))(()())())())()(()()(())()(()((()(())()((()(())(()))()))()))()()()((()(()()(()()(()))(() ((()()))(()((()()()))()()(())(())()()()))))())()()))))()(()()((()(()))(()()(())()()((()))(())))) ((()))(()()()())()()()(()(())()(())()()))))(()((())()((()()(())(()))))()()())()(()))))(()())) (((((((((()))))))))()(())(((()(()(())()))(()))(()))))())))(((()))(()()()))()(()))()((((()())()() (()()))))(())()(((())()))(()))(((()(()))))))(()))(()))))()))))(()))()((()))))()(())(((())()(()(())) ()((()))(()((())()))(()))((()((())((()()(()))(())))))))((()(()))()))(()((()))((((()))))()()() ((())))))()())((()())()))()(())()((((())))()(()))((()((()))(()()))()(()()))()()((()(()()(()() ()))))(())())))(())((()(()((()())(())()()()))))(((()(()()(()))))(())(()(())((((()))()()((((()(()) ()()))((()((())(()))()))))))()()))()())())(())((((()))(()))()(())((((())(())()()(()()(()(())()()())))) ()()()((()))()(()))))()())()))()())(())()(()((())(()((()())()()((()(()())))))))()))(()))((((()) ()))()((()()()()(((()())())(((())())))(())()())(()))()()))())(()))(()((((((()()()))))()())()(()(() (())()))((()()()())()())())(((()(()()((()(()())()())()())()())()()())()()()((()()(((())(((()())()()) (()))((((()()()())()(())()))()))((((()(()(()))()(()((()()((()()()()()(((())()()(((((()))()))))())) (())(()))((()))(())(()))(()((()))))()())())())))()))()()((())((()()(()))(())((()()())()()() (((((()(()))())()()())()(()))()()))((()))()))((()()(()(((()()((())()()()(()(())(()((()())))((((()))) ()(()(())()(()((())()))))()(()((()(()()(()(()))))(()()))(())()))()))))()");			

	Test	Expected	Got	
✔	`cout << mininumBracketAdd(")()())()))()()()((((((()()()))())(()(()(())))))()))())(()())())) ()))((()()((()((((((()()())()(())))))(()))((()(((()((()))()))))()()))()((())))(()))((()()()))() (())()))()))()((()()))((((((()())()))()))((()()))()((((((((((()(()(()(()())()))))()(()()))))()()))()) ()))((((())(()))()()()(())()(((()))))))))())))((((((((((()()))))()()))())))()())))))))))(()))(() (()))()))()(()()))()))()))())))()()())(()()())(()()(((((()()))))()))()())()()))((()((()))()()))(())))) ((((((()(()()())((((()))()()()((()()()))()))(()))()))(()()()()))))((()))))))))(((()()))))))))))()()) ((())(()))()))((()(()))(()()()())(())()))((()())()))()((((())))((()()((()()())()((()()))()()(()()((() ())((()()))((())((()))(()()((()(()))((()())()))(()))(((()())()()))(()(()))(()(()))()()))((((((()(() (((()))()()()(()()((()())()))(()()))()()(()()))(())))))()))()((()((()(()))(()))))))()))))()))()))((((() (((())((((()()()()(())(()))()(())())(()((((()(()()))())(()(()(()))())())(()()()))))(()(()))(())) (((())(()((((((()())()))((((()())))())))))))(())()())()())((()))(()()()))((()()))))()())((())()() ())((()())((())()))(()))))))((()))((((()))(()))((((((()())))()(()()))()(()((())()(())()())()()() ())((()))))())))()()()())))()()()))))()()()))((())()((())()()()))))))()))))(()))()()(((()())()() ())()()((((((()(()))((())((()(()()))(()))()()(()()()))())()((())((()(()((()())(()((((()())) ((((()((()())()())()()))()(()())()(()))()()((((((()()))(()(())()())))()()()(()(((()(((((()()))(() ()()((((()()))))()()(((((((()(()()(()()(())())(())()())(()(())))())()()(()())()(((((()((()((()(() ((((((()((()((()))()())(()((()(()()))(()))()(())()((((()((((()((((()(()((())()(((((()(((()(() (()))(()((())(()(()())(()()(()()))))(()((((((()(())((()))()))((())()()(()(((()((()(()()()())()((() (())))))))))()())))((()(()))())()((())))((()()))((()))()())())(()())()()())))))()((()())))()) (()(((()((())()))((()((())))(()(()))(((()()()()()((()(()())(())(((()((()))()()))) (((((()))))))))))()((()))((()))())((()()))()(())((()(()))(()()(()()(((())())))(((())()() ((((((()()()())))()))))()(()((()))()(()()(()))((()))((()))())))()((((((()(()))((((()()()(((()() ()))()())))))()((()()()()((()))())(())(()())()((()))(()()()(()))(()()()(()))()()()()))))())((())) ())))(((((())(((()(()))(((()()())()()(((((((((((()()))(((()()()(((()()))))()((((()()()()()())()) ())(()()())(()(()(()())))(()((((()((((()()))(((()((()))))()((()(((()()()()(()()()))()(()))()(()))))(((((()()()()))))()())(()()())()(()())(()))(()()))()()()(()(((()()))))))))(())))))))))))) (()(()(()()(())))(()((((((((()(()()))))))))(()()(()))((()))(((()))(()(()()))(()))()(()()())())()))(())))) (((()((()))))()()))())(()())(()()(()())(()()(()()((()()))())(()())((()())(()()))()()))(()())(()()) ((((()((()))()))()()())((()))()())(()()())((((()((((()))(()())(()()(())))))))(()))))(()))((()() ()()())((()()()()))()))())(()))()))(()()))())()))()((()()((((()(()()()))))((((((((()())(()()))(()))) ()((((()(((()()(((((((()()(((((((((()((((()))(((((((()(((()(()()(()(()())))))()()))))()()()()() (()))))(()((()())(()))((())()))(()))()(()((()()))())(()(()((()())(()(()())))(()()())()))()) (()()()))()()()))())((()((()()))()))()(((()()))()((()(()((((()))()()(()()))(((()))))((((((()))(()() ()(()))())(()))()()(((()))(()())(()())()((()()()(()()(((((()()((()((()()()()()((((()()(((()())) (()))((((())))))(())())(((((((()()((()()(()()(((()())(()()((((()((((())()()(()((((((()()()(())) (())(((()))(()))(()()(()()))()((((()())((()()()))()(((((()))))))))(()())(()()())()())()() ()))))())(()))(((()(()(()())(()))()(()()))((())()(()(()))((()))))(()()())))(((()())((())))(()(()) ()(()())()(()()(()))()((((())))()(((()()))()()((()(()(((()()))()))()((((()))())(()()(()(((()) ((((((((()())((((()((((()))(()((((((()()()(()((((((()()(((())())))(()()))()(()()))(())(((() ())(((((()(((((()(())))()))(()))(()))((((()(((((()(()()((((())(((((((()()((()))))(()()(()())() ((()))))(((((())()))((((()()))(((())()()) ...snip...)())(((()(()()))(()(()))()))(()()(()()) ()((()()))(()))((())))))()())(()()())(((()(()())()((((()()()))((()(()(()()))()()(()())()(((((()(()))(()(()))())))())))()(()()())))((()()((())(((())())(()(())))))()()))()((((())))()(()) ())()()))(()))()))(())((()()((()()()()())()()()()(()()()((()()()()()()()())()()()()())()()()()((()) (())()())((())((((((((())))(()((())((((((((()()))))(((()(()))))()()((()))()())))()(((((())()))) (()(()()))()())()()))()))(()))((((()(()()((()(()))()))()()()))))()((()((())((())((()()()() ((((((())()((((((((()()((()))()()((()((((()((()()()((((())))))))))()()(()))()))))()()()((((()))(((((()))))(()()((()))()((()()()(()()(()()(()))())))))))(()()()))(()()((()())))()(((((()()() ()(((((()(()(((()((())((((((()((()()(()()))()()((()()()(()()()()()(()((()()(()()(()()()((()()))) ()(()()()())())))((()(((()())))))()())()))))())()))(()()()()()(()()()()()()()())()()))()()()()()() ((()()(()(((()())))()()()))((()))()))((()))))))()(()(()()((()())))()))))((()(()()()(()()())) ((()))))(()((()()((()()()(()(()())()))(()((((()()())()((((()()((((((()()(()(((())))()() ((()()))()()()()((()()))()))(()((()()))()()(()()(((((()()()(((()()(((()((())()()(((()()))()(() ((((((()((()()((((((()(()((()()()()()()(((()((((()))((((((()()()())))(()))((()))((()()(()())))))) ()()()()(((()))))()()))(()))())(()(()()())()(()()))))()(()()()()(()()))())))()()(() ()(((()()()(((())))))(())()((()())))(()()(()()))))))(()))((())))()()))))))(((()()(())))()())) ())))()()(((())(()()))))))()(()))))())())(()))((((()(()()()(())(()((((((()))((((((())))() ()))(())))))()))()()()()))((()()(((()()((())((()((()()))))((())()()()((()))((()()))()))))))) ((((()()((((((((((())))()()))))))()()()()()()()())()()())()((((((((()((()()((((((((()()(() (()(()()(())))()())((()))))((()()))))()()))()())))((()()()()(())((()((()()((()))(()))()()) (()))(()())())()((()()(()()((((()((((()))()()()(())))()()()())()(()()((((((()((())()() ((((((()()(((()()((())(()))))()()))))))()))()())))))((()()(()))))))()()()(()((((()()(()()()((()())(()))))))(((((((((()()((((()()()()()(()))()((((((()()((()))))()))()())()()()()(()(((()((((())()()))))((((())()()())))((())(()))()()(((((()(()()(()()((((()()(((()(()()()(((()))))))))) ()()(()))))()()(()))))(())()()()()(()(()))((()))()(()()()))((()(((((()()(()()()()()))))))()(((()`	202	202	✔

	Test	Expected	Got	
	((((()())()))))((()()(()())()((()()(())())((()()))((((())()))())(()(()))))(()((((()()(((())())(()(((()) (()(()))))()()((()()(())(()))(())()()))()(())()((((()(())(((((()(())(((())((((()))()(())(((((() ((((()(()))))()(()))((()()()(())()))()(()(((())()())))()))()()))()()))))))()())(()((((()((())()() (())(()(())((()((()())((((()()((()()()()()())(()))(())((())(()))())()()())()(()))()((()))()) (())())())())()(()((()(()()((())))())((()()(()())()((()))()))()((()()())((()(())(()()))))(((()) (((()())()))((((()()((()))())(()(()()))(()))(((((()))(()()()((())(()(())(()(())(()())))())())(() ()))()()))))()(())()(()()))()())()(()))))())(()))()())()()(()()()((()(()(())((())()((()(()())) ((((((()()(()(()())(((()((()()()))()(()((()))(())((()(()(()()((()(((())(()(((()()))()())((()()()) ())))))((())()))))()))()()())((((()))(())()()))((((()()()()()(()))((())((())(()))((()))())))() (())((()))()))))())())(()(()()()((()()()(()()(()))()(((((()((((()(()((()))))())))((((()(() ((((((()))())()))((((())))))()))()((()()()))()())(()))()((()(())()(((()(()(())()()(()())()(((())() (((((()()()((((())(()))(())(())())((((()))()((()))()))())))()())()((((()(()(()))()))(())()() (())))()))(()())()((()()()((()(()(((((()))()))()((((())(()()()((((())()()))))))())))()((((((()(((()) (((((((()())((())()()))()(((((((())(()()(())(()(()(((())()))(()((()))(()((((())()(((((())())))))((()())) ((())())(()()((())((((())(()((((())(((()(()))(()()))))))))((((()(((()))))()(((((((()(()(()((((((()))())))) ()(((()()))((((()(()()))))))(())()(((()((())())(()))((()((()))())))())((())(()(((()((((((");			

<table>
<tr><th></th><th>Test</th><th>Expected</th><th>Got</th><th></th></tr>
<tr><td>✔</td><td>cout << mininumBracketAdd("(()))(((()()((()()())()))()(((()(()))((((()(())())()((((((()(()))))(()()
((((()()())(()())()((((()(()(())(())(()()()(((())(())())())((((((((())())(()))))(((())()()(()()))))()(()
(())()))(((((())()()))()))(()())()())()))()()()()()()()())((((())))))()())(((()((()((((((())()())))()
()))))()())()((()()))))()))((())((()))()))))((()()()((()())))))()((()()((()(()()((((()()(()()(((()()
(()))()))((()()()()()()))()))(())))()))))()))((((()()((((()()()((((()(())))))))))(((()))()))(()))))))
(()))))()))(()()()()()(())))))()()))()()((()()((()())(()))((((())))(()))))()((()(()))()()(()())()())
((((())()))()((())))))())))((()))(()()()())()()))(())(()()((()()((()(()())()()()))()(()())()()()()((((
())))))(())((()))()))()))(((((()(()((()(((())(()()))))()()(()()))))(((()()(()()()())(()))))()(()))()
()()((())(()(()())())()())(())(()((()()))(()()()())((()()(()))()(()))))((((()()(()()(()))))(((()()()()((((
())))))((()())()))((()()))(())((()))(()))())()))()((()()((()()))))(()))(()())(()))((()()))(((()((
((()((((()()((((())((()()((()(()()((((()()()())()))(())((()())())((()((((((())())((((()())(())))))))(())()
(()(()))((((()))((((()))(())())(()(((())))()))(()(()))(()))(((()(((())((()))(((((((())
(())))))())())()))(()))))(((()()))))(())((((((()((((()()()))))(()))))()()))(((()()(())())))())
()))))))))()()()()()()()(()()()(()()(()()()()()(())())())())(())(()())(()))(()))))))())))((((((((()
(())(())(()))))))()))(())()()()((()(((((((((()(()))))()()()))(())(())(()))(()))))((((())))))()
((((())((()()(()()((((((()(((()(())))))())()))()()(((()(()))()))))))((((()()()))())((()())((((()()))
())()()(((()(())()()((((()(()(())(())((()()()))((()))))(()(()()))))((((()))(((()(((((()()(())(())((()))
()()()))()(())())()()())()()(((())())()))()()()((((()())(()())()())())()))))())(((((()((((()((((()
()())()()))()((())()))()))())(()))(()()())(()(()()((((()((())())(()))())(())()())())()(())()(())(()
((())(())()()()((((()))())()(()())((()))((((((()((((()()())()((((((()(((()(())(())())())()()))
(((((())((((()()(()()(()))()(())()()(()()()()()((()(()))(()))()((()())())()))())))()(())()())
((()))))())))((((((((()(()(()))))((()(()((((()(())(())()))()()((()()()())((((()((()))(()))(()()
()))))(())()((()((()))())()())()()(()(()))(()(()()(()(((((()((((((()(((((()()))(()()(()))()())()
((()(((((((()(()()(())((()(())()()((((((((()()))()))()))()()(()()()((((()())()()()()()(()))()))(()((()
((())())(()))()(()()(())(()())))))))))))))((((()(()))(()(())(()(()))()()()(((()(()))(((((()())()
()()(())((()))(()(()(()))()()()))(()))(())))(((()))(()(()(((()))(()()))()()(()()))))()()(((()
(()))(((((((((()())(())((((()()))()))))(()(((()))))(((())))(()))))((()()()())()())(())()()(((()
(())))))))(()((((())()())())(()(()()()(())))()()()()()()(((()()(()()(())())(((()()(((())))
((())((((())))()())(((((()((()(()()()((((()()((()()(())))(()()(()))(()))()()()()()(()))(((()()
(()(()())(()))(()(()))))((((()(()))))(()()()(((())(()()()()()()()()()()((((()())))))()())
((()((()()()))))()()))(((()))(()(()(()))())()(()(((()(())(()(()(()))(())())()(()(()())))(()())(((())()
()(()()((()))()()()()))((()))()((()(()()()()))()))()((((()))()(((()()((((()))))())()((()())))()())))(((()
(()))((((()((()))(((((((()(()(())(()))()((()))(()))))(()((((()(()()()())))()()()()()()(((()))()
())))(())()(()(((()))()()()())()((((((((()()))(())))))((())))()((()((((()(()()()))))))()()(((((()
(()))()(()(()()(()))))))())()()())(()()(()((((()(()))(((()))))(()))))())()())()(())(())()(())(())
((())(())(())()()(()()))((()()))(()()(()()((()((((()(()()())()(()))))(()(()()(())()())((((()())
())))()()))))())(())())((((()(((((()))))()(())(()(()()))))(()(()))()()((((((())()))()))
(())(())()))))(()(()()(()())(())((((()))(()())()(()())(()))())((()(()()))(()((((((()))()(((()((
(()()())()(())(())())()(()()())(()))())(()((()()))(()))()(()(()()(()()()()))
())))))()(()(()(()))()((((())))())))()())((()((((((((())()(()))))())(()()))((()(()()))()(())))))))(((()
((()()))()())(((((()(()()(()))()))()))()))(()((((((...snip... ()()()((()(())()))())))((())(((()()())
()((())()))(())(()()))()))())))())))(()()()))(())()(())((((()()(()()()))(())()()(())(()()(()
(()()()())()((())()()))))))(())((())()((((((((()))((((()(()))()()((()()((((()((((()())))))(()(())
()()(()))(())(())()())((()))()((()()()())))))()(())()()(())(()()))()((((()()(())()())())(()()))))()()
((((()((()))((((()))))))()()))()()())))()()))()(()))()(((((()((()()()))))())()(()))))(()()))()()()))))
((())))()(()((()((()))((((()(())()(()(())()((((((()()()(()())(())((()()))(()()())(())(()()()()()
((()((()))())(()()(())()((()(()()))))((((()(((())()))(((()()(())())()()((((((((((())((()()(())())
()))))()))(((((()()())()((((()(()))()(()(()))(()())()(()())(()))()))()()(()()((((())))))())((((())
()))))(((((()(()(()()))()((()))())(()()()()()((()))((())()(()((((()))))()()(()())((())((()()))()(((()
())())((())((((()(((()))()()()))()((((()((()((((((()()())())))))()()(()())(()())(((()())())(()())
()()()((((((())()))()()(()()))))(((()))(())()(())(((()))())((((()(()()(()))(((()()))())((((())((((((()
((()))()(())())(())(()())(()))(((()))()())(((((((()(()()(())()())()))))(()(()(((((()(()((()))((((())))))
((()))()())()())))))(((()()()())()((((()((((())())()())(()((((()()()()())()(()())(((()()(()(((((()))
(())(()()(()())()(()))((()))()(()))()(((((((((((())))))((((()((()(((())))((()))())())(((()()
(())((()(()(())(())())(()))))(((())(((())((((()(())))(())()(())()(()(((()(((((()(())())((()(((())))()
((()))()(()))(()(()()))((((()())((((()())()))())))))(((()))(()((()))()())((((((((((()()())(())())()())
(())())()(()())))))()))(()))()(()()())))(()))()()(((()()()(((()()())))((((()(()()()))(((()(()()())
((((((()))((((((()(()())(())))())))(())(((())(())(()))(())(())()(()))(())))())(())(())))())())(()())
()())((()())())(((((((((()((()()())()))))()(()))(((()()()(((((((()(((()()()())(())((((())()((())())))))
(()()((())((((()()())()))((()(()((())))()()((()))(()(()())((()))))()()())((()))(())()()((((()()())
()()))()())()((((()())()())(()))(()))))())))))()(()))()()))((()()())(((((()((()))(()()(((((())))))
())))))))))((()(()))(())(((()(()(()()((((()(()()))((()()))((((()())))()()(()()))))(())()((())
((((())()))((()())())((((((()(()(()()))(()))()))(())(()((((((())(((()())))((((()))()(())()()(()))((()
())))))()()(()))))()()))((()(()(()()))())))))()()))))()()))()(()))()()))(())))()()(((()(()))
()(()()))()((()))())))(())()(()()))))()()))))()()()()))()(((()))()((((((((()))((((()()()())()(()))()</td><td>426</td><td>426</td><td>✔</td></tr>
</table>

	Test	Expected	Got	
	()())()))()))(()()()(()((((())()()()))(()()()()))))()))()()((()(((()(((()))((((()((((()((())())))) ())((())()(()(()((((())())(())(())((())())(())()()(()(((((())(())())((()()()()(()((((())(()(((())))() (()))()))()(((((((())()(())(()))(((((())(()))(()(()(((()(((()((((((()))((())(())((((()))(()))()()(()))))(() (())(()))()())(()()))(()))()()()(()())(((()))(((())(())(()))((()())()()((()(()))()))))(())((((())) ()))()))))))))()(()())((((()((((()(((()(()()))()()((()(()()(()()(()((((()()(()()(()()(()()()))(((()))))() ()(()()()((()))((((((()))())(()((()((())))()))()(()()((()())(()((((()((()(((()((())()()))))))() (())))()(())()(()((((())()))(()())((()((())(())((()(()(()((())())(()()(()(())())(()(((((())(()(())))) (())))(()(()()())((()()(()))(()(())()()))))()((()(((((())()()(())()(()((())()(())(((((((()))())(()) ((()((((((())((((()))())))()))()()((()((()(((()()(())))))())((()()(())(()))()(())()(()(()()())()))))(()() ()())()())(()))))(()()(()))(()((()))())((()))())(()()(()()())(((((()(()))(((()((()))(()(())))) (()))))()(()))))())((())()))())()(()))))())(()))()())()((())((())((()))))))()((((())((()(()()() (()))())()()((())((())()))((())((()((()(()(()()(()())()(()()((())()(())((((((()()()()((((()()()()()()) ((((((()((((()(()))((())(()(()())((())((()())(((())((())(())(((())((()((()))(()))))(())(())())(()))(()) ((((((()()(()))())(()))((()((()((()(()((()((((()))()()())()(()())((()((((()()(()()))())))))())(() ())((())))))))))(((()()(()((())(()((()((((()((()()))))()((())))())((()(()(((()()())()()(((()))(()()((() ())(()))))(()()()(()))()((((()((()))())))))))))((())()))((()(()())(");